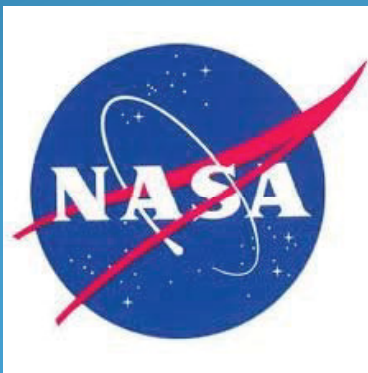


Libraries on the Go Mobile Websites and Apps

Dudee Chiang, Information Science Specialist
Alexander Smith, Information Science Specialist
Jet Propulsion Laboratory, California Institute of Technology

Research Center Information Managers (RCIM), May 5, 2011



Copyright 2011, California Institute of Technology
Government Sponsorship acknowledged



Outline

- Trends on going mobile
- Mobile web sites vs. apps
- Examples from publishers/societies
- JPL mobile website

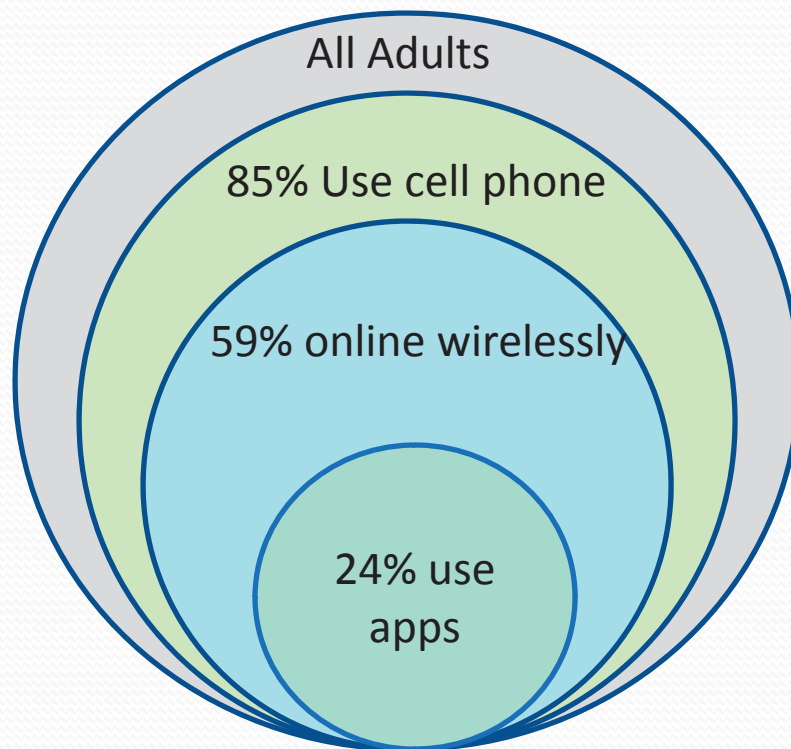


Trends

“Go mobile or get left behind”

- 25% waking hours on mobile devices
- 63% information management functions doing nothing to support handheld devices
- “Top 10 Information Management Predictions for 2011”, Outsell, Dec 2010

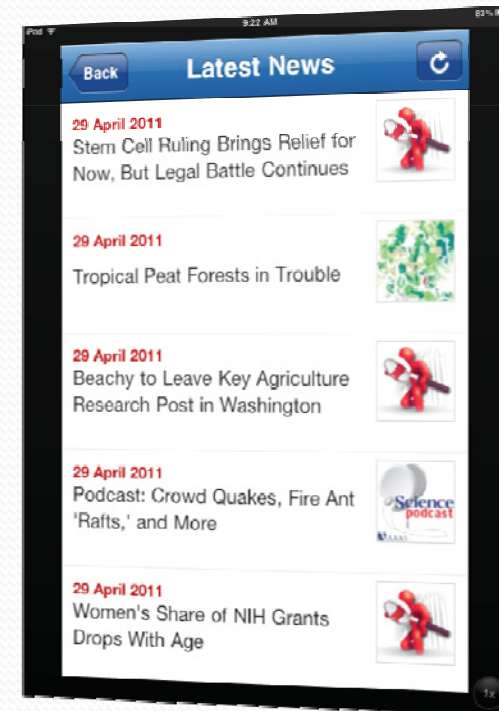
❖ More and more people are using mobile devices to access the Internet



- Of all adults (18+), 85% use a cell phone
 - By ages:
 - 18-29: 89%
 - 30-49: 69%
 - 50-64: 49%
 - 59% online wirelessly
 - Including laptops
 - 24% use apps
- Pew Internet and American Life Project, Reported March 2011



- Longer, richer media on one page/site for laptop and desktop computers



- Short, context-based, specific interactions on mobiles (m-web, apps);



High Level Considerations

Mobile web sites

- Available to all devices
- Server detects requesting device and displays mobile site
- Simple navigation
- Point back to full site

Apps

- Device specific (iPhone, iPad, Droid, etc.)
- Distributed via devices' store or market
- One App on multiple devices:
 - Kindle for iPad, PC, Mac, etc.
- One App One Purpose/Function
 - YouTube, games

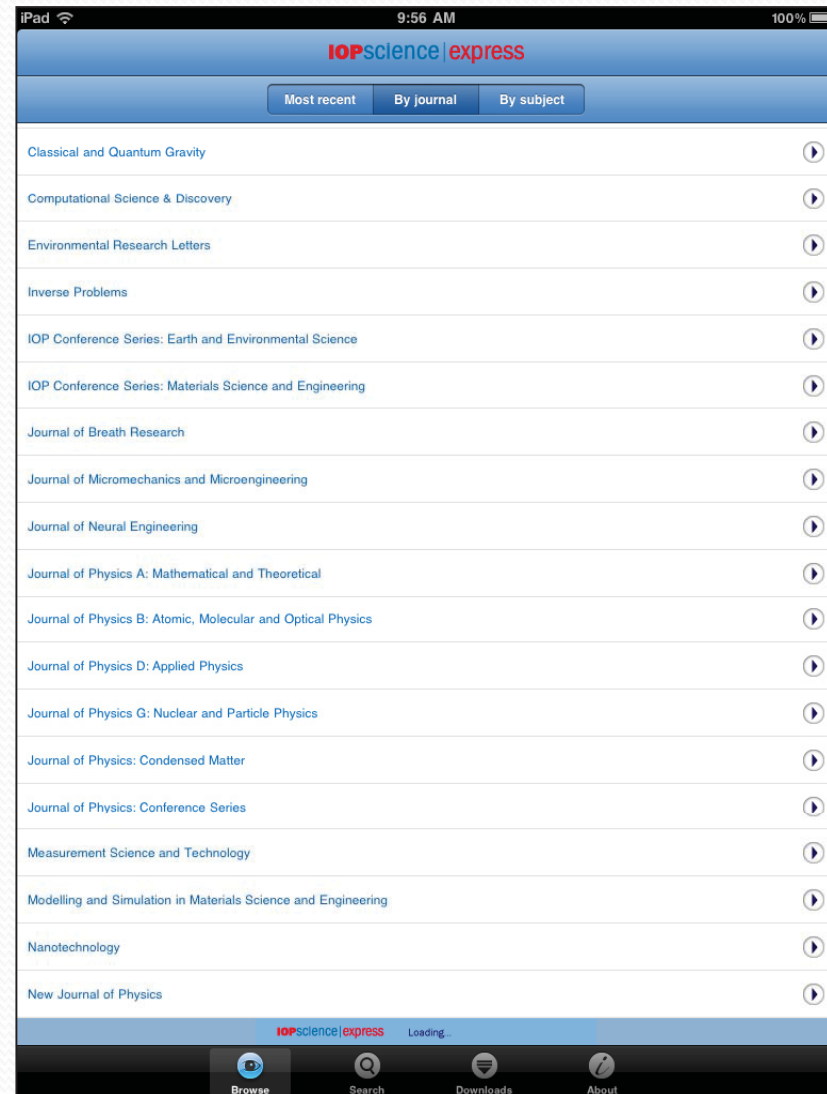


What Do Our Suppliers Have?

- Learn from their examples
- Information Professionals as trusted experts and curators of apps

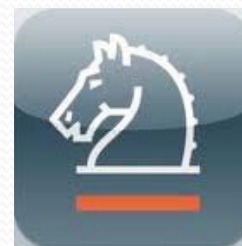
Institute of Physics (IOP) app

- Free app
- 20 full text articles per month for anyone, no institutional subscriptions required!



“Library Friendly” apps

- Free apps
- Free browse, search, abstracts
- Need institutional subscription to download full text
- ACS app costs \$4.99 to download
- Science selected free articles



Need Individual Subscriptions

- App may be developed by 3rd party vendor

- Harvard Business Reviews:

HBR Reads: free app, individual subscription to HBR

HBR Today: \$2.99 app, access to the HBR web site, no scholarly articles

HBR Tips, HBR Stat: free apps, free short articles refer to full articles in HBR



Mobile web sites

- IEEE Xplore

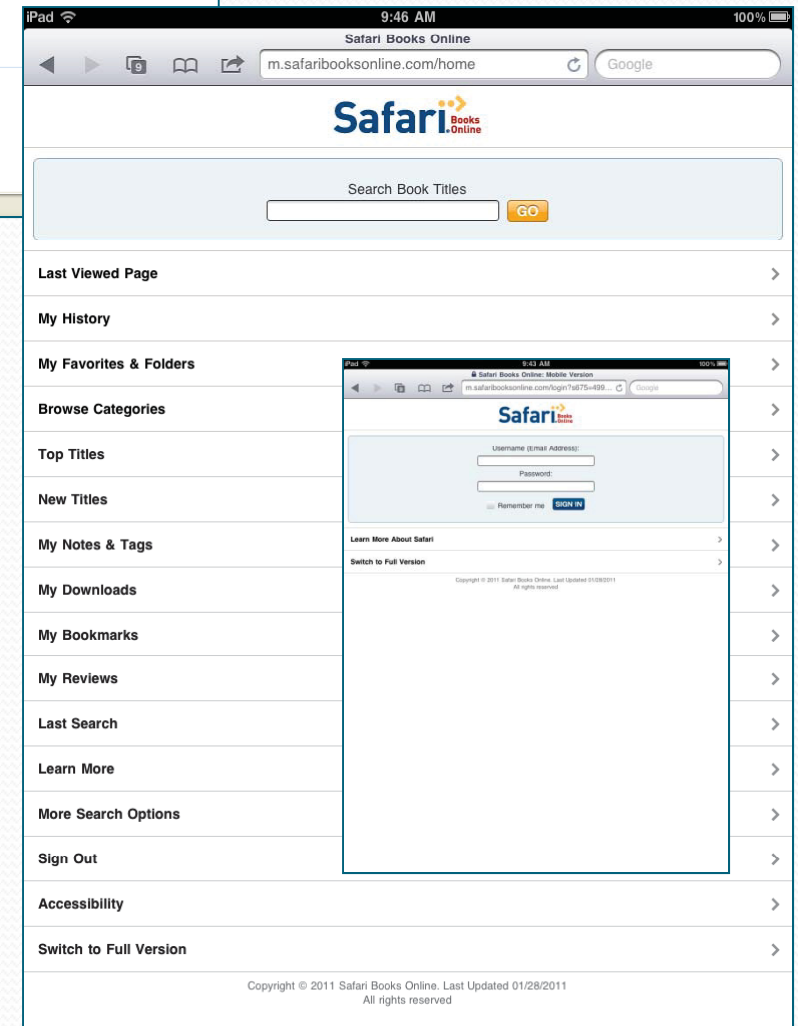
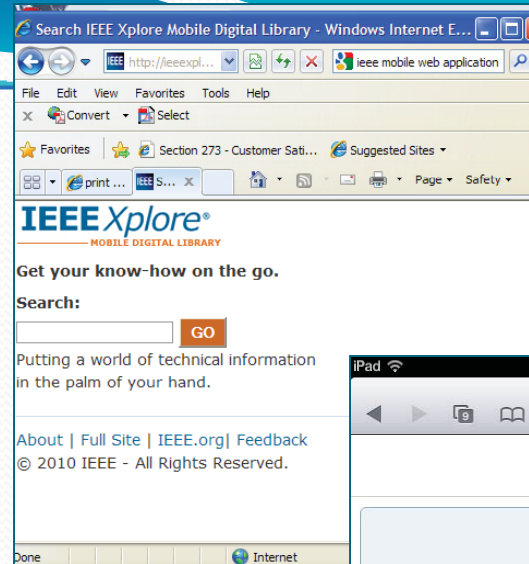
<http://ieeexplore.ieee.org/mobile/>

Search, browse, email references by anyone

Need subscription for full text

- Safari Books Online

<http://m.safaribooksonline.com/login>



BEACON Library Mobile Site

Goals:

- Lightweight / Efficient
- Reuse targeted pieces of content from WordPress in an elegant way
- Harvest our ebooks list from delicious.com and re-use content in the mobile site
- Integrate a basic authenticated ILLiad 'My eDeliveries' feature
- Utilize few or no images, make layouts/styles dependent upon modern stylesheets
- Integrate Historical Photo of the Month & blog features in an elegant fashion.



WordPress is the BEACON site's content management system (CMS). It makes it simple to serve up content independent of layout or style. Perfect storm for mobile web.



A lightweight HTML shell is presented to the user by WordPress based on device type. Once the HTML is successfully loaded, jQuery is set up to act on the data by loading images & content. Content is not loaded all at once, only pieces on-demand, which increases efficiency. So for instance: when I click "News & Events" jQuery requests data only associated with News & Events.



jQuery looks first in an internally stored database of content that is cached from initial requests to wordpress. If the data is expired past it's custom expiration policy, fresh data is reloaded from Word-Press and stored locally.



The Problem Surrounding HTML5 Local Storage

For mobile developers the landscape was severely fragmented when it came to a new protocol: HTML5 and it's new feature of local storage.

HTML5 brought local storage into the picture but HOW it was queried and stored was a whole new ballgame.

Storing Frequently Used Information was tedious and not possible for small teams wearing many hats or teams without a dedicated mobile developer. Each browser handles local storage differently and has its own unique syntax for storing and retrieving data.



localStorage

```
// Save data to a the current session's store  
localStorage.setItem("username", "John");
```

```
// Access some stored data  
alert( "username = " +  
localStorage.getItem("username"));
```

Drawbacks: No iteration over multiple objects with a common name. No deserialization for objects.



SQLite

```
// first create a table with a schema  
CREATE TABLE 'atablename'(key INTEGER PRIMARY KEY ASC, temp, a, b);  
CREATE TABLE 'joinedtable'(key INTEGER PRIMARY KEY ASC, foreignid, c,d,e,f,etc);  
//now insert some data in the 1st table  
Query(INSERT INTO atablename  
VALUES(", 100, 'b', 1));  
//prepare to do our 2nd insert into table#2  
Var theglue = last_insert_rowid();  
//more data insertion  
Query(INSERT INTO joinedtable  
VALUES(", theglue, etc, etc));  
//now we can finally, slowly select some data. The more JOINS = slower times.  
Var result = query(SELECT * FROM atablename JOIN joinedtable ON atablename.key = joinedtable.foreignkey WHERE .....)  
//not done yet, lets loop through the data...  
for(var i=0; i<result.length;i++){  
$( 'div#temperature' ).html( result[i].temp );  
}
```

Drawbacks: Schemas, more keystrokes, can be overkill for small projects, no deserialization for objects. Have to escape quotes.



DOMStorage

```
// Save data to a the current session's store  
sessionStorage.setItem("username", "John");
```

```
// Access some stored data  
alert( "username = " +  
sessionStorage.getItem("username"));
```

Drawbacks: No iteration over multiple objects with a common name. No deserialization for objects.



A New Perspective on Cross-Platform Storage/Apps

Lawnchair is something new to the game that aims to tie together the various local storage methods with a common query syntax that gracefully wraps around the native storage methods.

Lawnchair is maintained by Brian Leroux who also maintains PhoneGap.

PhoneGap allows web developers to write a native smartphone app in HTML and JavaScript and publish it across many device formats (ie: the same code makes the same app for Android, iPhone, Palm, BlackBerry, etc).

HTML is very good at buttons, layouts, and data presentation, thus making a perfect platform for mobile apps.

The solution to the silo'd environment was using Lawnchair.js (an open source project). We are able to set caching policies for image data and page content. For instance, news and events items are refreshed only if the content has not been refreshed in 3 days. Using Lawnchair brought the total possible page size down from 150 Kilobytes to 51 Kilobytes!

```
Make new datastore:  
var lc = new Lawnchair('name');  
  
Save document  
lc.save({val1:'abc',val2:234,somethin  
gelse:'111abcd'});  
  
Find document  
lc.find('r.val1 == "abc",function(r){  
  //iterate over many  
  var dynamicvariable = r.val2;  
});  
  
//remove document  
lc.remove({key:'unique_key'});
```

